

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1 1. (Currently amended) A method for facilitating typesafe software design
2 while supporting structured composition of a software system, comprising:
3 receiving a first invocation of the software system;
4 assigning a first context to the first invocation;
5 examining the first invocation to locate components of the first invocation;
6 registering a unique factory to build each component, wherein these
7 factories are registered using the first context; and
8 when a component is needed, building the component using the unique-a
9 factory associated with the component, whereby building the component after
10 each component has a registered factory eliminates potential problems with
11 initialization circularity.

1 2. (Original) The method of claim 1, further comprising:
2 receiving a second invocation of the software system;
3 assigning a second context to the second invocation;
4 examining the second invocation to locate components of the second
5 invocation;
6 registering a unique factory to build each component, wherein these
7 factories are registered using the second context; and
8 when a component is needed, building the component using a factory
9 associated with the component, whereby building the component after each

10 component has a registered factory eliminates problems with initialization
11 circularity.

1 3. (Original) The method of claim 2, wherein components from the second
2 invocation are not available to the first invocation.

1 4. (Original) The method of claim 1, further comprising providing an
2 additional factory for an extended component of the first invocation.

1 5. (Original) The method of claim 1, wherein registering the unique
2 factory to build each component involves placing a key and a related factory
3 identifier into a storage structure.

1 6. (Original) The method of claim 5, wherein building the component
2 using the factory associated with the component involves using the key to retrieve
3 the related factory identifier from the storage structure.

1 7. (Original) The method of claim 6, wherein the storage structure includes
2 a hash table.

1 8. (Currently amended) A computer-readable storage medium storing
2 instructions that when executed by a computer cause the computer to perform a
3 method for facilitating typesafe software design while supporting structured
4 composition of a software system, wherein the computer-readable storage medium
5 includes magnetic and optical storage devices, disk drives, magnetic tape, CDs
6 (compact discs), and DVDs (digital versatile discs or digital video discs), the
7 method comprising:
8 receiving a first invocation of the software system;

9 assigning a first context to the first invocation;
10 examining the first invocation to locate components of the first invocation;
11 registering a unique factory to build each component, wherein these
12 factories are registered using the first context; and
13 when a component is needed, building the component using the unique-a
14 factory associated with the component, whereby building the component after
15 each component has a registered factory eliminates potential problems with
16 initialization circularity.

1 9. (Original) The computer-readable storage medium of claim 8, the
2 method further comprising:
3 receiving a second invocation of the software system;
4 assigning a second context to the second invocation;
5 examining the second invocation to locate components of the second
6 invocation;
7 registering a unique factory to build each component, wherein these
8 factories are registered using the second context; and
9 when a component is needed, building the component using a factory
10 associated with the component, whereby building the component after each
11 component has a registered factory eliminates problems with initialization
12 circularity.

1 10. (Original) The computer-readable storage medium of claim 9, wherein
2 components from the second invocation are not available to the first invocation.

1 11. (Original) The computer-readable storage medium of claim 8, the
2 method further comprising providing an additional factory for an extended
3 component of the first invocation.

1 12. (Original) The computer-readable storage medium of claim 8, wherein
2 registering the unique factory to build each component involves placing a key and
3 a related factory identifier into a storage structure.

1 13. (Original) The computer-readable storage medium of claim 12,
2 wherein building the component using the factory associated with the component
3 involves using the key to retrieve the related factory identifier from the storage
4 structure.

1 14. (Original) The computer-readable storage medium of claim 13,
2 wherein the storage structure includes a hash table.

1 15-21 (Canceled).